On the group theoretical background of assigning stepwise mutations onto phylogenies

Fischer, Mareike; Klaere, Steffen; Minh Anh Thi Nguyen, [No Value]; von Haeseler, Arndt

Published in:
Algorithms for molecular biology

DOI:
10.1186/1748-7188-7-36

ALGORITHMS FOR MOLECULAR BIOLOGY

RESEARCH Open Access

On the group theoretical background of assigning stepwise mutations onto phylogenies

Mareike Fischer[1,4], Steffen Klaere[2*], Minh Anh Thi Nguyen[3,4] and Arndt von Haeseler[4]

Abstract

Recently one step mutation matrices were introduced to model the impact of substitutions on arbitrary branches of a phylogenetic tree on an alignment site. This concept works nicely for the four-state nucleotide alphabet and provides an efficient procedure conjectured to compute the minimal number of substitutions needed to transform one alignment site into another. The present paper delivers a proof of the validity of this algorithm. Moreover, we provide several mathematical insights into the generalization of the OSM matrix to multi-state alphabets. The construction of the OSM matrix is only possible if the matrices representing the substitution types acting on the character states and the identity matrix form a commutative group with respect to matrix multiplication. We illustrate this approach by looking at Abelian groups over twenty states and critically discuss their biological usefulness when investigating amino acids.

Keywords: Maximum likelihood, Maximum parsimony, Substitution model, Tree reconstruction, Group theory

Background

Alignments of homologous sequences provide fundamental materials to the reconstruction of phylogenetic trees and many other sequence-based analyses (see, e.g., [1,2]). Each alignment column (site) consists of character states that are assumed to have evolved from a common ancestral state by means of substitutions. Any combination of the character states in the aligned sequences at one alignment column represents a so-called *character* [3], which is sometimes also called *site pattern* [4]. Given a phylogenetic tree and an alignment that evolved along the tree, Klaere et al. [5] showed, for binary alphabets, how a character changes into another character if a substitution occurs on an arbitrary branch of the tree. The impact of such a substitution is summarized by the so-called *One Step Mutation* (OSM) matrix. The OSM matrix allows for analytical formulae to compute the posterior probability distribution of the number of substitutions on a given tree that give rise to a character [5].

*Correspondence: steffen.klaere@gmail.com
[2]Department of Statistics and School of Biological Sciences, University of Auckland, Private Bag 92019, Auckland, New Zealand
Full list of author information is available at the end of the article

Nguyen et al. [4] extended the concept of the OSM matrix to the four-state nucleotide alphabet while developing a method, the MISFITS algorithm, to evaluate the goodness of fit between models and data in phylogenetic inference. There, the OSM matrix is constructed based on the Kimura three parameter (K3ST) substitution model [6]. Nguyen et al. [4] illustrated how one can apply the Fitch algorithm [7] to compute the minimal number of substitutions required to change one character into another character under the OSM setting. In the present paper, we deliver a proof of the validity of this algorithm.

In addition, the OSM matrix can be constructed only if the matrices representing the substitutions, the so-called *substitution matrices,* and the identity matrix form a commutative or Abelian group (see, e.g., [8]) with respect to matrix multiplication [4]. The link between Abelian groups in phylogenetic models has been studied before, most notably by Hendy et al. [9]. Further, an extension of nucleotide substitution models with an underlying Abelian group to joint states at the leaves of a tree has also been studied by other authors. Bashford et al. [10] introduced an approach very similar to OSM to study the multi-taxon tensor space. Bryant [11]

also introduced a very similar framework to study the Hadamard transform of [12] in the light of multi-taxon processes.

In this work, we first introduce standard phylogenetic notation. We then formalize the construction of the OSM matrix, and which part of its construction is used in the MISFITS algorithm. We further present possible extensions of the OSM framework to arbitrary alphabets. We will show that the MISFITS algorithm in fact computes the minimal number of substitutions needed to change one character into another character. Moreover, we discuss the extension of the algorithm to substitution models which do not have an underlying Abelian group. Finally, we discuss the Abelian groups available for amino acids.

Notation and problem recapitulation

Notation

Recall that a *rooted binary phylogenetic X-tree* is a tree $\mathcal{T} = (V(\mathcal{T}), E(\mathcal{T}))$ with the following properties: There is one vertex $\rho \in V(\mathcal{T})$ with indegree 0 and outdegree 1, which is called the *root* of $\mathcal{T}$. All edges $e \in E(\mathcal{T})$ are directed away from ρ, and all vertices $v \in V(\mathcal{T}) \setminus \{\rho\}$ have indegree 1 and outdegree 0 or 2. Vertices with outdegree 0 are usually referred to as *leaves* of $\mathcal{T}$. Remember that for an X-tree, there are exactly $|X| = n$ leaves, which is why there is a bijection between the set of leaves of $\mathcal{T}$ and the taxon set X. Thus, when there is no ambiguity, we use the terms *leaf* and *taxon* synonymously. Moreover, we often just write "phylogenetic tree" or "tree" when referring to a rooted binary phylogenetic tree.

Furthermore, recall that a *character* f is a function $f : X \to \mathcal{C}$ for some set $\mathcal{C} := \{c_1, c_2, c_3, \ldots, c_r\}$ of r *character states* ($r \in \mathbb{N}$). We denote by $\mathcal{C}^n$ the set of all r^n possible characters on $\mathcal{C}$ and n taxa. For instance, for the four-state DNA alphabet, $\mathcal{C}_{DNA} = \{A, G, C, T\}$ and the set $\mathcal{C}^n_{DNA}$ consists of all 4^n possible characters.

An *extension* of f to $V(\mathcal{T})$ is a map $g : V(\mathcal{T}) \to \mathcal{C}$ such that $g(i) = f(i)$ for all i in X. For such an extension g of f, we denote by $l_{\mathcal{T}}(g)$ the number of edges $e = \{u, v\}$ in $\mathcal{T}$ on which a substitution occurs, i.e. where $g(u) \neq g(v)$. The *parsimony score* of f on $\mathcal{T}$, denoted by $l_{\mathcal{T}}(f)$, is obtained by minimizing $l_{\mathcal{T}}(g)$ over all possible extensions g. Given a tree $\mathcal{T}$ and a character f on the same taxon set, one can easily calculate the parsimony score of f on $\mathcal{T}$ with the famous Fitch algorithm [7]. Moreover, when a character state changes along one edge of the tree, we refer to this state change as *substitution* or *mutation*. As for our purposes only so-called manifest mutations are relevant, i.e. those mutations that can be observed and are not reversed, we do not distinguish between mutations and substitutions, which is why we use these terms synonymously.

Construction of the OSM matrix

We now introduce the OSM framework in a stepwise fashion. The aim of the OSM approach is to determine the effects a single mutation occurring on a rooted tree $\mathcal{T}$ has on a character evolving on that tree.

The first task of this approach is to formalize the term mutation and its effects on a single character state in $\mathcal{C}$. A mutation is an operation $\sigma : \mathcal{C} \to \mathcal{C}$ which is bijective, i.e. it satisfies the following condition:

C1. For all $c_i \in \mathcal{C}$ there is a $c_j \in \mathcal{C}$ such that $\sigma(c_i) = c_j$, and if $\sigma(c_i) = \sigma(c_j)$, then $c_i = c_j$.

This guarantees that a mutation affects a character state in a unique fashion. It is well-known that any bijective function on a finite discrete state set is a permutation (e.g., [13]). Thus, a mutation is a specific instance of a permutation applied to a character.

The next step is to select the set Σ of admissible permutations acting on $\mathcal{C}$. It is mathematically convenient to select Σ such that it forms an Abelian group [9] with a regular (transitive and free) action on $\mathcal{C}$. Hence, Σ satisfies the following conditions:

C2. For every pair $c_i, c_j \in \mathcal{C}$ there is exactly one permutation $\sigma \in \Sigma$ such that $\sigma(c_i) = c_j$, i.e., the action of Σ on $\mathcal{C}$ is regular.

C3. For all $\sigma_1, \sigma_2 \in \Sigma$ also the product $\sigma_1 \circ \sigma_2 \in \Sigma$. Mathematically speaking, Σ is closed with respect to concatenation of its permutations.

C4. For all $\sigma_1, \sigma_2 \in \Sigma$ we have $\sigma_1 \circ \sigma_2 = \sigma_2 \circ \sigma_1$. Thus, Σ is commutative, and hence the order in which we assign permutations is irrelevant for the outcome.

C5. There is an element $\sigma_0 \in \Sigma$ such that for all $\sigma_1 \in \Sigma$ we have $\sigma_1 \circ \sigma_0 = \sigma_0 \circ \sigma_1 = \sigma_1$, i.e. there exists a so-called neutral element, namely the identity, in Σ. For all $c_i \in \mathcal{C}$ only $\sigma_0(c_i) = c_i$, i.e. σ_i is fixed point free for all $\sigma_i \neq \sigma_0$.

C6. For every $\sigma_1 \in \Sigma$ there exists a $\sigma_2 \in \Sigma$ such that $\sigma_1 \circ \sigma_2 = \sigma_0$. Mathematically speaking, for every element of Σ there exists an inverse element. This guarantees that every permutation can be reversed within a single step.

C7. For all $\sigma_1, \sigma_2, \sigma_3 \in \Sigma$ we have $\sigma_1 \circ (\sigma_2 \circ \sigma_3) = (\sigma_1 \circ \sigma_2) \circ \sigma_3 = \sigma_1 \circ \sigma_2 \circ \sigma_3$, i.e. the associative law holds.

It should be noted that any set of permutations is associative, i.e. satifies C7. Thus, for a set of permutations Σ to be Abelian with a regular action on $\mathcal{C}$ it only needs to satisfy C1−C6.

In the following, we consider the matrix representation of permutations. A permutation matrix over $\mathcal{C}$ is an $r \times r$ matrix such that $\sigma_{c_i c_j} = 1$ if $\sigma(c_i) = c_j$, and 0 otherwise. We consider it equivalent to discuss a permutation

or its corresponding matrix. Therefore, concatenation "$\circ$" is equivalent to the matrix multiplication "$\cdot$". We use σ to denote a permutation or a permutation matrix, depending on the context.

Example 1. *In genetics, the most commonly used character state set is $\mathcal{C}_{DNA} = \{A, G, C, T\}$. There are two different Abelian groups for four states, namely the Klein-Four-group $\mathbb{Z}_2 \times \mathbb{Z}_2$ and the cyclic group $\mathbb{Z}_4$. The Klein-Four-group is constructed from the cyclic group $\mathbb{Z}_2$ over two elements, the identity τ_0 and the flip τ_1. These take the matrix form*

$$\tau_0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \tau_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

The Klein-Four-group consists of the four Kronecker products of these two matrices, i.e. $s_0 = \tau_0 \otimes \tau_0$, $s_1 = \tau_1 \otimes \tau_0$, $s_2 = \tau_0 \otimes \tau_1$, and $s_3 = \tau_1 \otimes \tau_1$. The Kronecker products here yield 4×4 matrices, e.g.,

$$s_1 = \tau_1 \otimes \tau_0 = \begin{pmatrix} 0 & \tau_0 \\ \tau_0 & 0 \end{pmatrix} = \begin{array}{c} \\ A \\ C \\ G \\ T \end{array} \begin{array}{c} \begin{array}{cccc} A & C & G & T \end{array} \\ \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \end{array}.$$

The set $\Sigma_{K3ST} := \{s_0, s_1, s_2, s_3\}$ coincides with the substitution matrices under the Kimura 3ST model [6]. In particular, s_1 describes transitions within purines (A, G) and pyrimidines (C, T), s_2 represents transversions within pairs (A, C) and (G, T), and s_3 represents the remaining set of transversions within pairs (A, T) and (C, G).

The second Abelian group over four states, the cyclic group $\mathbb{Z}_4$, is formed by selecting a 4-cycle, e.g., $A \to G \to T \to C \to A$ and concatenating this cycle with itself. The resulting set of permutations $\Sigma_{\mathbb{Z}_4}$ contains the following elements:

$$s_1' = \begin{array}{c} \\ A \\ C \\ G \\ T \end{array} \begin{array}{c} \begin{array}{cccc} A & C & G & T \end{array} \\ \begin{pmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix} \end{array},$$

$$s_2' = {s_1'}^2 = s_1' \cdot s_1', \quad s_3' = {s_1'}^3, \quad s_0' = {s_1'}^4.$$

Note that there are actually six different four-cycles for $\mathcal{C}_{DNA}$. These result in three distinguishable Abelian groups. Bryant [14] generates his cyclic group with the four-cycle $A \to C \to G \to T \to A$, and shows that the resulting set Σ_{K2ST} underlies the Kimura 2ST model [15], where s_2' corresponds to the transition within purines and pyrimidines, and s_1' and s_3' are the (not further distinguished) transversions.

The next step in constructing the OSM matrix is to construct a set $\Sigma^{\mathcal{T}}$ of operations over $\mathcal{C}^n$ governed by $\mathcal{T}$, and based on the permutation set Σ. To this end, we first define Σ^n as a set of operations which work elementwise, i.e. for $f = (f_1, \ldots, f_n) \in \mathcal{C}^n$ and $\sigma \in \Sigma^n$ we have

$$\sigma(f) := (\sigma_1(f_1), \ldots, \sigma_n(f_n)), \quad \sigma_i \in \Sigma.$$

This can also be described by the Kronecker product, i.e. equally

$$\sigma(f) = \sigma_1 \otimes \cdots \otimes \sigma_n(f). \tag{1}$$

This means that there are r^n different operators in $\Sigma^n = \Sigma \otimes \cdots \otimes \Sigma$.

Remark 1. *Therefore, for any pair of characters $f, g \in \mathcal{C}^n$ we can find an operation $\sigma \in \Sigma^n$ such that $\sigma(f) = g$.*

Another noteworthy consequence of using the Kronecker product is that the elements of Σ^n are permutations over $\mathcal{C}^n$ [16,17], and in fact Σ^n satisfies our Conditions C1−C7, i.e. Σ^n is an Abelian group over $\mathcal{C}^n$.

In the OSM framework we assume that the permutations acting on a character $f \in \mathcal{C}^n$ are derived from the underlying rooted tree $\mathcal{T}$. If permutation $\sigma_i \in \Sigma$ acts on the pendant edge leading to taxon $j \in X$, then the associated permutation matrix $\sigma^{j,i}$ acting on $\mathcal{C}^n$ has the form

$$\sigma^{j,i} := \bigotimes_{l=1}^{j-1} \sigma_0 \otimes \sigma_i \otimes \bigotimes_{l=j+1}^{n} \sigma_0.$$

If a permutation acts on an interior edge e, then it simultaneously acts on the states of all descendant taxa of e, i.e. all those taxa whose path to the root passes e. E.g., assume Taxa 1 and 2 form a cherry, i.e. their most recent common ancestor, 12, has no other descendants, and permutation $\sigma_i \in \Sigma$, $i = 1, \ldots, r-1$ is acting on the edge leading to this ancestor. Then, we get the permutation

$$\sigma^{12,i} := \sigma_i \otimes \sigma_i \otimes \sigma_0 \cdots \otimes \sigma_0 = \sigma^{1,i} \cdot \sigma^{2,i}. \tag{2}$$

This shows in particular that a Kronecker product of some permutations acting on each character state is equivalent to the matrix product of the permutations acting on the entire character. The right hand side equation

shows that a single permutation on an internal edge has the same effect as simultaneously applying the same permutation on the pendant edges of all descendant taxa. In other words, if $\mathrm{de}(e)$ denotes the set of descendants of edge e, and $\sigma_i \in \Sigma$, then

$$\sigma^{e,i} = \prod_{j \in \mathrm{de}(e)} \sigma^{j,i}. \tag{3}$$

Note that the set Σ^X of all permutations acting on the pendant edges is a generator of Σ^n, i.e. the closure of Σ^X contains all permutations in Σ^n. Since Σ^n contains a single permutation to transform character $f \in \mathcal{C}^n$ into $g \in \mathcal{C}^n$, and since Σ^X generates Σ^n, there is a shortest chain of permutations in Σ^X which transforms f into g. Σ^X is also the set of permutations implied by the star tree for X. In general, the set of all permutations on tree $\mathcal{T}$ is

$$\Sigma^{\mathcal{T}} = \left\{\sigma^{e,i} : e \in E(\mathcal{T}),\ i \in \{0, \ldots, r-1\}\right\},$$

where r is the number of states in Σ.

For every X-tree $\mathcal{T}$ we have $\Sigma^{\mathcal{T}} \supseteq \Sigma^X$, and therefore $\Sigma^{\mathcal{T}}$ is a generator for Σ^n, too. An illustration of such a generator set $\Sigma^{\mathcal{T}}$ over the character set $\mathcal{C}^n$ is the so-called *Cayley graph* [18], which has as vertices the characters of $\mathcal{C}^n$, and two characters $f, g \in \mathcal{C}^n$ are connected if there is a permutation $\sigma \in \Sigma^{\mathcal{T}}$ such that $\sigma(f) = g$. In [5] Cayley graphs have been presented as alternative illustrations of the tree $\mathcal{T}$ over a binary state set $\mathcal{C} = \{0, 1\}$.

Example 2. *Regard the K3ST model from Example 1 and the rooted two-taxon tree depicted in Figure 1a. With this* $\Sigma^{\mathcal{T}}_{K3ST}$ *is given by the set*

$$\begin{array}{lll} s^{e_1,1} := s_1 \otimes s_0, & s^{e_2,1} := s_0 \otimes s_1, & s^{e_{12},1} := s_1 \otimes s_1, \\ s^{e_1,2} := s_2 \otimes s_0, & s^{e_2,2} := s_0 \otimes s_2, & s^{e_{12},2} := s_2 \otimes s_2, \\ s^{e_1,3} := s_3 \otimes s_0, & s^{e_2,3} := s_0 \otimes s_3, & s^{e_{12},3} := s_3 \otimes s_3. \end{array}$$

Each permutation which acts on the characters is thus a symmetric 16×16 *permutation matrix depicting a transition* ($s^{e,1}$)*, transversion 1* ($s^{e,2}$)*, or transversion 2* ($s^{e,3}$) *along edge* $e \in E(\mathcal{T})$*. Figures 1b-d display the permutation matrices for a transition on branch* e_1 ($s^{e_1,1}$)*,* e_2 ($s^{e_2,1}$) *and* e_{12} ($s^{e_{12},1}$)*, respectively. Figure 1e shows the Cayley graph associated with* $\Sigma^{\mathcal{T}}_{K3ST}$*.*

We are now in a position to recall the definition of the *OSM matrix* $M_{\mathcal{T}}$ for a rooted binary phylogenetic tree $\mathcal{T}$ as explained in [5] and [19]. For an edge $e \in E(\mathcal{T})$ we denote by p_e the relative branch length of e, i.e. its actual branch length (expected number of substitutions per site) divided by the length of $\mathcal{T}$ (the sum of all branch lengths). Thus, one can view p_e as the probability that a mutation is observed at edge e assuming that a single mutation occurred on $\mathcal{T}$. Clearly, $\sum_{e \in E(\mathcal{T})} p_e = 1$. Further, denote by $\alpha_{e,i}$ the probability that this mutation on e is of type $i \in \{1, \ldots, r-1\}$ with $\sum_{i \in 1}^{r-1} \alpha_{e,i} = 1$ for all $e \in E(\mathcal{T})$. Then the OSM matrix is the convex sum of the elements in $\Sigma^{\mathcal{T}}$, where each permutation $\sigma^{e,i}$ is multiplied by $\alpha_{e,i} p_e$, the probability of hitting the edge e with permutation $\sigma_i \in \Sigma$. Thus, we obtain:

$$M_{\mathcal{T}} = \sum_{e \in E(\mathcal{T})} \sum_{i=1}^{r-1} \alpha_{e,i} p_e \sigma^{e,i}. \tag{4}$$

$M_{\mathcal{T}}$ can be regarded as the weighted exchangeability matrix for all characters under the K3ST model assuming that a single substitution occurs on the tree $\mathcal{T}$. Figure 1f depicts the OSM matrix for the tree in Figure 1a. Here, colors indicate relative branch lengths p_e, and patterns denote permutation types α_i. E.g., a blue square with horizontal lines indicates the product $p_{e_2}\alpha_{e_2,1}$, i.e. the probability of observing a transition s_1 on edge e_2.

The transformation problem

With the construction of $\Sigma^{\mathcal{T}}$ we have generated the tools needed to formally describe the computations in Step 4 of the MISFITS algorithm [4]. Given a rooted tree $\mathcal{T}$ and two characters f and f^d in $\mathcal{C}^n$, we want to compute the minimal number of substitutions required on the tree to convert f into f^d. [4] presented an efficient procedure to compute this minimal number of substitutions.

Algorithm 1

INPUT: rooted binary phylogenetic tree $\mathcal{T}$ on leaf set X, characters f and f^d on X, Abelian group Σ.
STEP 1: Using Remark 1, find the substitution type σ_i which translates f_j into f^d_j for all positions $j = 1, \ldots, |X|$. Let $\sigma \in \Sigma^n$ be the resulting operation, i.e. $\sigma(f) = f^d$.
STEP 2: Let $c := c_1 \ldots c_1$ be a constant character on X with $c_1 \in \mathcal{C}$. Let $h := \sigma(c)$.
STEP 3: Calculate $m := l_{\mathcal{T}}(h)$.
OUTPUT: m.

We prove the correctness of our algorithm. In our framework, m corresponds to the minimum number of permutations $\sigma_1, \ldots, \sigma_m \in \Sigma$ such that $\sigma_1 \otimes \cdots \otimes \sigma_m(f) = f^d$. In this form, m has multiple equivalent interpretations. It is the length of the shortest path between f and f^d in the Cayley graph for $\Sigma^{\mathcal{T}}$, where this path corresponds to $\sigma_1 \otimes \cdots \otimes \sigma_m$. Further, m corresponds to the minimum power (k) of $M_{\mathcal{T}}$ such that $M^j_{\mathcal{T}}(f, f^d) = 0$ for $j < k$ and $M^k_{\mathcal{T}}(f, f^d) > 0$, because a positive entry in $M^k_{\mathcal{T}}$ means that there is a concatenation of k permutations connecting the associated characters.

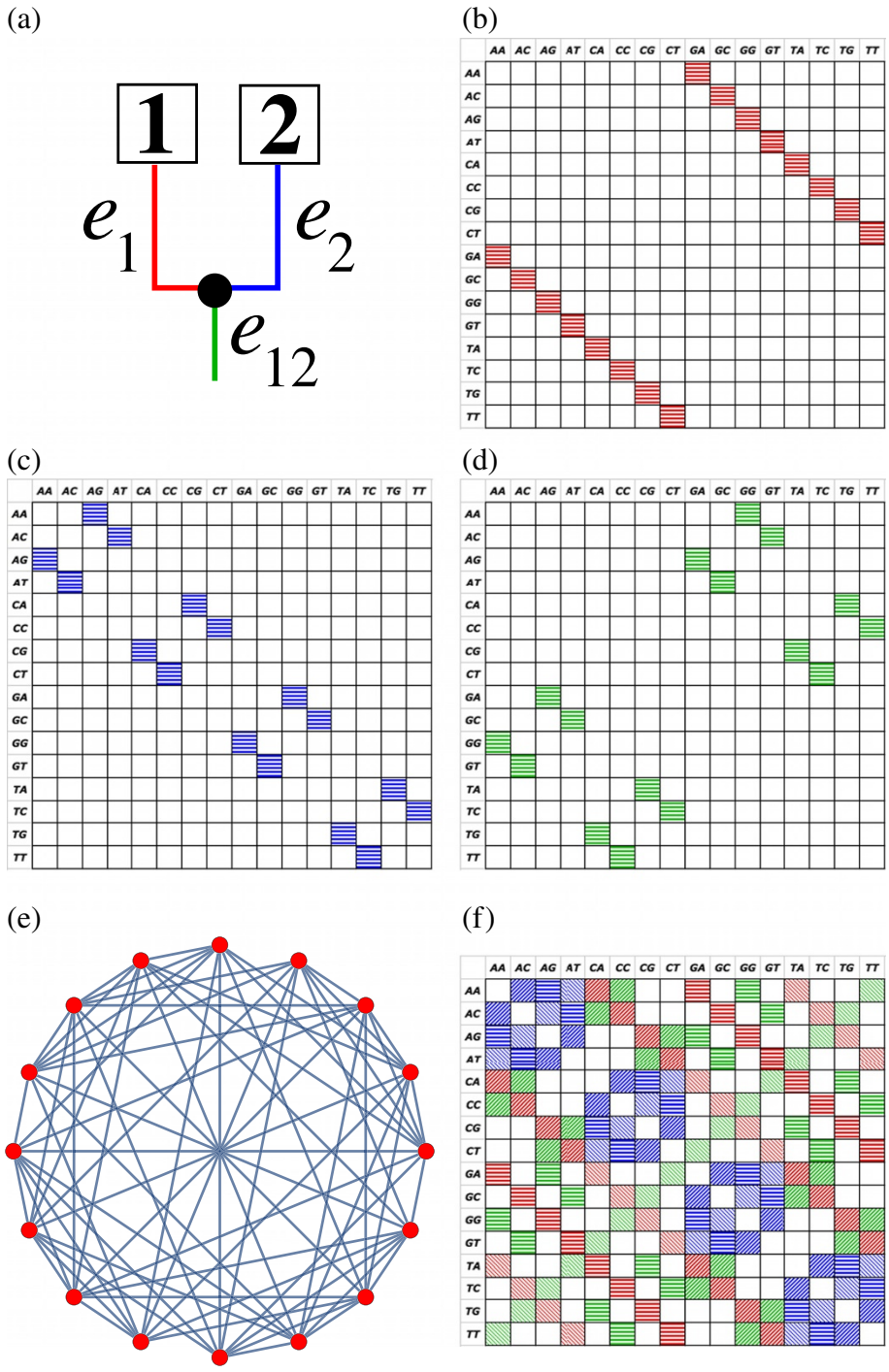

Figure 1 Construction of the OSM matrix. (**a**) A rooted tree with taxa 1 and 2. (**b**) A transition s_1 on the left branch e_1 (the red branch) changes a character into exactly one new character as depicted by the red horizontal stripe cells of the permutation matrix σ^{e_1,s_1}. The matrix has 16 rows and 16 columns representing the possible characters for the alignment of two nucleotide sequences. The permutation matrices generated by s_1 for the right branch e_2 (blue) and for the branch leading to the "root" e_{12} (green) are displayed in (**c**) and (**d**), respectively. The corresponding Cayley graph for the tree is illustrated in (**e**). The convex sum of all the weighted (by the relative branch length and the probability of the substitution type) permutation matrices generated by all substitution types for all branches is the OSM matrix of the tree ($M_{\mathcal{T}}$) as shown in (**f**). Horizontal stripe cells represent the probability of the transition s_1; diagonal stripes the transversion s_2; and thin reverse diagonal stripes the transversion s_3. The colors of these cells indicate the relative branch lengths and follow the colors of the branches as in (**a**). Thus, these colors also depict the branch origin of the substitutions.

Example 3. *Figure 2 demonstrates how Algorithm 4 works under the K3ST model, i.e. when the group is* $\Sigma = \Sigma_{K3ST}$ *(Figure 2a). Consider the rooted five-taxon tree in Figure 2b and the character GTAGA at the leaves. Assume that the character GTAGA is to be converted into character ACCTC. By comparing the two characters position-wise, we need a substitution* s_1 *on the external branch leading to taxon 1 to convert G into A at the first position. Similarly, we need a substitution* s_1 *on the external branch leading to taxon 2, and a substitution* s_2 *on every external branch leading to taxa 3, 4, and 5. Thus, the operation* $s := (s_1, s_1, s_2, s_2, s_2)$ *transfers the character GTAGA into the character ACCTC. As the operation s also translates the constant character AAAAA into GGCCC, converting GTAGA into ACCTC is equivalent to evolving the character state A at the root along the tree to obtain the character GGCCC at the leaves. The Fitch algorithm [7] applied to the character GGCCC with the constraint that the character state at the root is A produces a unique most parsimonious solution of two substitutions as depicted by Figure 2c.*

Results

The impact of parsimony on the estimation of substitutions

In this section, we provide some mathematical insights into the role of maximum parsimony in the estimation of the number of substitutions needed to convert a character into another one as explained above. In particular, we deliver a proof for Algorithm 4.

Theorem 1. *Let* $\mathcal{T}$ *be a rooted binary phylogenetic tree on taxon set X and let f be a character that evolved on* $\mathcal{T}$ *due to some evolutionary model and let* f^d *be another character on X. Then, the minimum number of substitutions to be put on* $\mathcal{T}$ *which change the evolution of f in such a way that* f^d *is generated can be calculated with Algorithm 4.*

Proof. Let $f, f^d, X, \mathcal{T}$ and Σ be as required for the input of Algorithm 4. Then, as defined in the algorithm, we have $\widehat{\sigma}(f) = (\widehat{\sigma}_1(f_1), \widehat{\sigma}_2(f_2), \ldots, \widehat{\sigma}_n(f_n)) = f^d$, where $\widehat{\sigma}_j \in \Sigma$ refers to the substitution type needed to translate f_j into f_j^d.

Considering the underlying tree $\mathcal{T}$, we may assume $\widehat{\sigma}_1, \ldots, \widehat{\sigma}_n$ act on the pending branches leading to taxa $1, \ldots, n$, respectively.

Now we show that it is equivalent to consider $\widehat{\sigma}(c)$, where c is a constant character, instead of $\widehat{\sigma}(f)$. Let $\mu \in \Sigma^{\mathcal{T}}$ be a transformation with $\mu(f) = f^d$. Then,

$$\widehat{\sigma}^{-1} \circ \mu(f) = \widehat{\sigma}^{-1}(f^d) = f. \tag{5}$$

Next, let $\widetilde{\sigma} \in \Sigma^{\mathcal{T}}$ be such that $\widetilde{\sigma}(c) = f$. Then, using (5), we have

$$\widehat{\sigma}^{-1} \circ \mu \circ \widetilde{\sigma}(c) = \widehat{\sigma}^{-1} \circ \mu(f) = f = \widetilde{\sigma}(c).$$

On the other hand, we can use the commutativity of the underlying Abelian group to derive

$$\widehat{\sigma}^{-1} \circ \mu \circ \widetilde{\sigma}(c) = \widetilde{\sigma} \circ \widehat{\sigma}^{-1} \circ \mu(c).$$

So altogether we have

$$\widehat{\sigma}^{-1} \circ \mu \circ \widetilde{\sigma}(c) = \widetilde{\sigma} \circ \widehat{\sigma}^{-1} \circ \mu(c) = \widetilde{\sigma}(c)$$

and therefore $\widehat{\sigma}^{-1} \circ \mu(c) = c$ and thus $\mu(c) = \widehat{\sigma}(c)$. As μ was arbitrarily chosen, this implies that any transformation which maps f to $\widehat{\sigma}(f) = f^d$ also maps c to $\widehat{\sigma}(c)$. Therefore, we have

$$\{\rho \in \Sigma^{\mathcal{T}} : \rho(f) = f^d\} = \{\rho \in \Sigma^{\mathcal{T}} : \rho(c) = \widehat{\sigma}(c)\}.$$

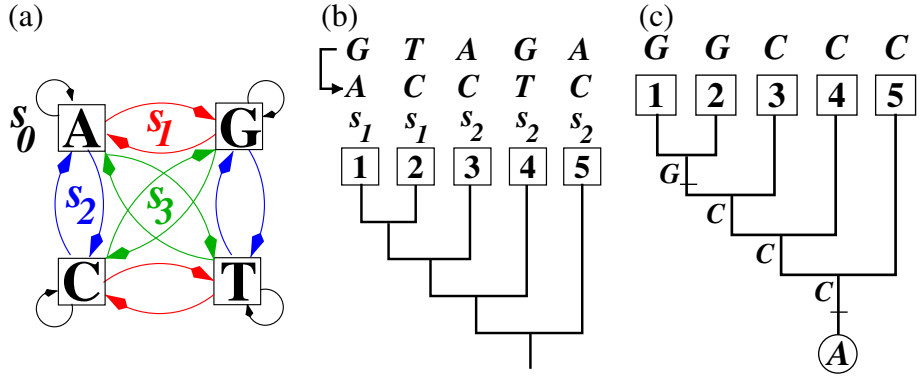

Figure 2 Computing the minimal number of substitutions to translate a character into another one. (**a**) depicts the Klein-four group Σ_{K3ST}, which consists of the identity s_0 and the three substitution types s_1, s_2, s_3 from the K3ST model. (**b**) In order to convert the character *GTAGA* into *ACCTC* under Σ_{K3ST}, we need to introduce the operation $s := (s_1, s_1, s_2, s_2, s_2)$. As the operation s also translates the constant character *AAAAA* to *GGCCC*, converting *GTAGA* into *ACCTC* is equivalent to evolving the character state *A* at the root along the tree to obtain the character *GGCCC* at the leaves. The Fitch algorithm applied to the latter produces a unique most parsimonious solution of two substitutions as depicted by (**c**).

The minimum number of substitutions to change f from f^d on $\mathcal{T}$ is just an element of the first set consisting of the fewest number of compositions. As the two sets are equal, we can investigate the second set rather than the first. So we need an element of the second set which consists of as few as possible compositions. Assuming that $\sigma = \sigma_1 \otimes \cdots \otimes \sigma_n$, we can assign $\sigma_1, \ldots, \sigma_n$ to the pending branches of $\mathcal{T}$ and treat them like character states to which we then apply the Fitch algorithm. This completes the proof. □

Informally speaking, the idea is as follows: As there is exactly one path from the root ρ to any taxon $x \in X$, we wish to determine whether we can 'pull up' some of the operations along this path in order to affect more than one taxon and still give the same result. This idea has been described above (Equations (2) and (3)), and it coincides precisely with the idea of the parsimony principle.

However, in order to avoid confusion regarding the operation σ as a character on which to apply parsimony, Algorithm 4 instead acts on the constant character. Clearly, in order to evolve the constant character $c := c_1 \cdots c_1$ on a tree with root state c_1, the corresponding operation would be $\widetilde{\sigma} := \sigma_0 \otimes \cdots \otimes \sigma_0$. Note that $\sigma(c) = h$ and $\sigma(f) = f^d$, and that two character states in h are identical if and only if the corresponding substitutions in σ are identical, too. Therefore, it is possible to let MP act on h rather than directly on σ.

By the definition of maximum parsimony, when applied to h on tree $\mathcal{T}$ with given root state c_1, it calculates the minimum number m of substitutions to explain h on $\mathcal{T}$. This number m is therefore precisely the number of substitutions needed to generate h on $\mathcal{T}$ rather than c. As $\sigma(f) = f^d$, m also is the number of substitutions needed to generate f^d from f on $\mathcal{T}$.

The impact of different groups

For any alphabet $\mathcal{C}$, there might be more than one Abelian group. Different groups might result in different numbers of substitutions required to translate a character into another character. We illustrate this observation using the following example.

Example 4. *Recall the starting point of Example 3, i.e. regard the five-taxon tree $\mathcal{T}$ from Figure 3b, and the characters f = GTAGA and f^d = ACCTC. Now, instead of using Σ_{K3ST} we use the permutations from the cyclic group $\Sigma_{\mathbb{Z}_4}$. In this setting, we need a substitution s'_3 (blue in Figure 3a) on the external edge leading to taxon 1 to convert G into A at the first position, and so on. Thus, we get the operation $s'_2 := (s'_3, s'_1, s'_3, s'_1, s'_3)$ such that $s'(f) = f^d$. We immediately see, that s' transforms the constant character c = AAAAA into h = CGCGC. The Fitch algorithm applied to the character CGCGC with the constraint that the character state at the root is A produces a unique most parsimonious solution of three substitutions as depicted by Figure 3c. Thus, under the Σ_c group we need one substitution more than under the Σ_{K3ST} group (cf. Example 3).*

Note that variation of the minimum number of substitutions needed to translate a character into another one between different groups is not surprising: As different substitution types are needed to translate one pattern into the other one, depending solely on the underlying group, one group might need the same substitution type for some neighboring branches in the tree and another group different ones. Informally speaking, this would imply that in the first case, the substitution could be "pulled up" by the Fitch

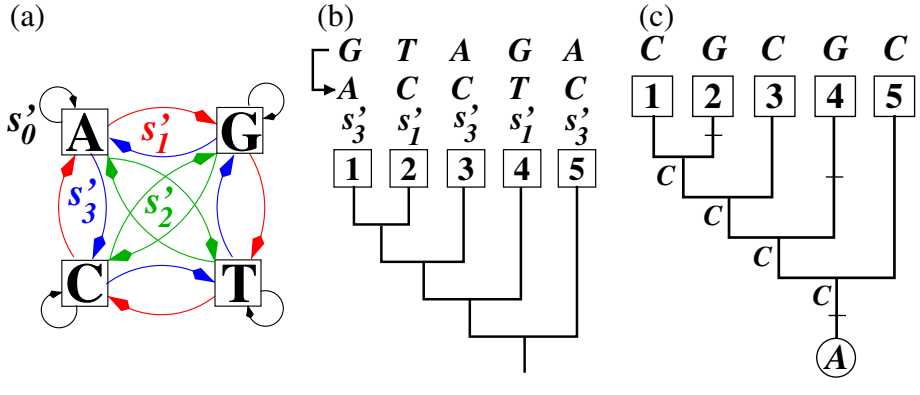

Figure 3 Converting one character into another character using the cyclic group. (**a**) depicts the cyclic group Σ_c, which consists of the identity $s'_0 \equiv s_0$ and the three substitution types s'_1, s'_2, s'_3 for nucleotide character states. (**b**) In order to convert the character *GTAGA* into *ACCTC* using this group, we need to introduce the operation $s' := (s'_3, s'_1, s'_3, s'_1, s'_3)$. As the operation s' also transforms the constant character *AAAAA* to *CGCGC*, converting *GTAGA* into *ACCTC* is equivalent to evolving the character state *A* at the root along the tree such that the character *CGCGC* is attained at the leaves. The Fitch algorithm applied to the latter produces a unique most parsimonious solution of three substitutions as depicted by (**c**).

algorithm to happen on an ancestral branch, whereas in the second case this would not be possible.

The link between substitution models and permutation matrices

In Examples 1 and 2 we have shown that the K3ST substitution model can be included into our framework. The connection between the Klein-Four-group and the K3ST model (as well as the one between the $\mathbb{Z}_2$ group and symmetric 2-state model) were described in-depth in [9]. This section aims at discussing alternative models and how to identify their use (or lack thereof) for our approach.

Most substitution models assume the independence of the different branches of a tree to compute the joint probability of the characters in $\mathcal{C}^n$. Therefore, they use the probabilities for substitutions among the character states in $\mathcal{C}$ along the edges of the tree $\mathcal{T}$. We now establish a probabilistic link between $\Sigma^{\mathcal{T}}$ and $\mathcal{C}^n$. This link is provided by Birkhoff's theorem:

Theorem 2 (Birkhoff's theorem, e.g., [20], Theorem 8.7.1)**.** *A matrix M is doubly stochastic, i.e., each column and each row of M sum to 1, if and only if for some $N < \infty$ there are permutation matrices $\sigma_1, \ldots, \sigma_N$ and positive scalars $\alpha_1, \ldots, \alpha_N \in [0,1]$ such that $\alpha_1 + \cdots + \alpha_n = 1$ and $M = \alpha_1\sigma_1 + \cdots + \alpha_N\sigma_N$.*

Therefore, the weighted sum of the permutation matrices in $\Sigma^{\mathcal{T}}$ yields a doubly stochastic matrix $M_{\mathcal{T}}$ as introduced above. $M_{\mathcal{T}}$ also describes a random walk on $\mathcal{C}^n$ governed by $\mathcal{T}$ where the single step in $\mathcal{C}^n$ is illustrated by the associated Cayley graph. Its stationary distribution is uniform, i.e. when we throw sufficiently many mutations on $\mathcal{T}$ then we expect to see each character with probability $1/r^n$.

Another, even more useful consequence of Birkhoff's theorem is the fact that it tells us which substitution models are suited for the OSM approach. If the transition matrix associated with the substitution model is doubly stochastic, then we find a set of permutations which give rise to the model.

Let us see how this influences the symmetric form of the general time reversible model (sGTR) with uniform stationary distribution. It has the transition probability matrix

$$P_{\text{sGTR}} = \begin{array}{c} \\ A \\ C \\ G \\ T \end{array} \begin{array}{c} \begin{array}{cccc} A & C & G & T \end{array} \\ \begin{pmatrix} 1-a-b-c & a & b & c \\ a & 1-a-d-e & d & e \\ b & d & 1-b-d-f & f \\ c & e & f & 1-c-e-f \end{pmatrix} \end{array}.$$

Assigning permutation matrices to the respective parameters yields the set Σ_{sGTR} with elements s_0 (identity) and

$$s_a = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad s_b = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$s_c = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}, \quad s_d = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

$$s_e = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, \quad s_f = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

The weighted sum of the non-identity elements yields

$$as_a + bs_b + cs_c + ds_d + es_e + fs_f = \begin{pmatrix} d+e+f & ab & c \\ a & b+c+fd & e \\ b & da+c+e & f \\ c & ef & a+b+d \end{pmatrix},$$

which is equal to P_{sGTR} if $a+b+c+d+e+f = 1$. Thus, the set Σ_{sGTR} is to sGTR what Σ_{K3ST} is to K3ST. However, Σ_{sGTR} does not satisfy condition C5, because $s_a, \cdots, s_f$ are not fixed point free. This can be seen as the main diagonal of $s_a, \cdots, s_f$ does not only contain zeros. It is also not commutative (condition C4) as e.g. $s_a \cdot s_c \neq s_c \cdot s_a$. And it is not closed under matrix multiplication (condition C3), which means that a concatenation of permutations in Σ_{sGTR} might lead to a new permutation not in Σ_{sGTR}, e.g., $s_a \cdot s_f \notin \Sigma_{\text{sGTR}}$. Other complex models like Tamura-Nei [21] do not even permit the decomposition of its transition matrix into the convex sum of permutation matrices. All of this shows why the overall applicability of complex models to the OSM approach is rather limited.

There are other approaches to describe phylogenetic models based on the group structure of their substitution matrices. In particular, Sumner et al. [22] use Lie algebra to construct OSM type matrices for the general Markov model, and discuss shortcomings of the group structure for the general GTR model [23].

Application to other biologically interesting sets

As stated above, OSM-type models require an underlying Abelian group. Thus, the OSM setting is applicable not only to binary data or four-state (DNA or RNA) data, but also to alphabets of 16 (doublets), 64 (codons), and 20 characters (amino acids) respectively. We compare such extensions to existing biologically motivated binning approaches and discuss their relevance.

As we have shown in the previous sections, the symmetric form of the Klein-Four-Group $\mathbb{Z}_2 \times \mathbb{Z}_2$ is mathematically beautiful, computationally convenient and biologically relevant. Similar statements can be made about all powers of $\mathbb{Z}^2$, including the biologically relevant alphabets of 16 (doublets) and 64 (codons) letters.

There are four Abelian groups for twenty-state alphabets, namely $\mathbb{Z}_2 \times \mathbb{Z}_2 \times \mathbb{Z}_5$, $\mathbb{Z}_4 \times \mathbb{Z}_5$, $\mathbb{Z}_2 \times \mathbb{Z}_{10}$, and the cyclic group $\mathbb{Z}_{20}$ (see e.g., [24] for a complete list of all groups with up to 35 elements). Their construction is analogous to the construction of the Klein-Four-group in Example 1. For example, the elements of $\mathbb{Z}_4 \times \mathbb{Z}_5$ are Kronecker products of one of the four permutations in the cyclic group $\mathbb{Z}_4$ with one of the five permutations of the cyclic group $\mathbb{Z}_5$.

Figure 4 shows a heat-map type visualization of an OSM-type matrix on a single-leaf tree where the coloring of the cells corresponds to the weights given to the 20 permutations in the respective groups. We see that the coloring pattern nicely reflects the four cosets of the subgroup $\mathbb{Z}_5$ in $\mathbb{Z}_2 \times \mathbb{Z}_2 \times \mathbb{Z}_5$. This can also be interpreted as a binning of the 20 states in the underlying alphabet into four sets of five elements each. If the weighting corresponds to a convex combination of operations, then the visualized matrix is doubly stochastic.

Binnings are also done for amino acids, using either biochemical properties or evolutionary divergence. An example of a biochemical binning is the hydrophobic index, where the 20 amino acids are binned into four groups, very hydrophobic, hydrophobic, neutral, and hydrophilic. Unfortunately, this binning does not correspond to any of the proposed Abelian groups. Moreover, it is difficult

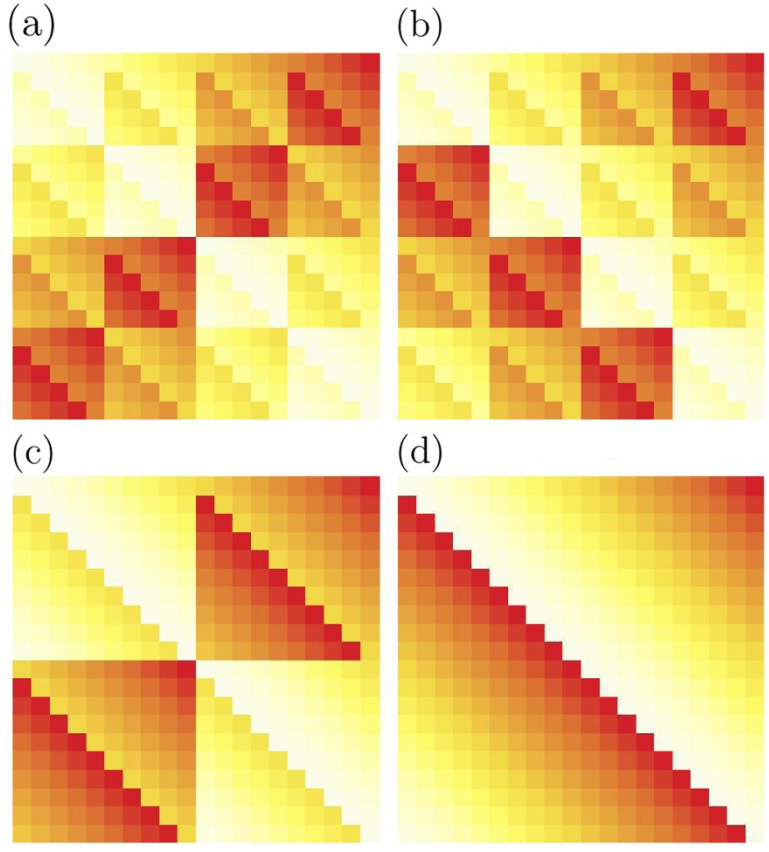

Figure 4 Matrices illustrate the four Abelian groups for a twenty-state alphabet. (**a**) the $\mathbb{Z}_2 \times \mathbb{Z}_2 \times \mathbb{Z}_5$ group, (**b**) $\mathbb{Z}_4 \times \mathbb{Z}_5$, (**c**) $\mathbb{Z}_2 \times \mathbb{Z}_{10}$, and (**d**) $\mathbb{Z}_{20}$. Each matrix visualizes the cosets of the subgroups of the depicted group and suggests an associated grouping of the 20 states.

to derive transitions between these groups just from the biochemical properties.

Transition matrices for evolutionary models for amino acid substitutions are usually generated by counting mutation types in the alignments (see, e.g., [25] for an overview). From these, optimal groupings can be obtained using clustering approaches [26]. The existence of estimates for the transition probability between all amino acids provides the possibility to get further information about between-group operations. These groupings could be forced to fit Abelian groups. However, as indicated in [26] a grouping into four groups of five amino acids each is rarely optimal.

Conclusions

In this paper, we provide the necessary mathematical background for the OSM setting which was introduced and used previously [4,19], but had not been analyzed mathematically for more than two character states. Moreover, the present paper also delivers new insight concerning the requirements for the OSM model to work: In fact, we were able to show that mathematically, it is sufficient to have an underlying Abelian group – which shows a generalization of the OSM concept that was believed to be impossible previously [4]. Therefore, we show that OSM is applicable to any number of states.

However, note that the original intuition of the authors in [4] was biologically motivated: The authors supposed that the group not only has to be Abelian, but also symmetric in the sense that each operation can be undone by being applied a second time. Thinking about DNA, for instance, this works: For example, the transition from A to G can be reverted by another substitution of the same type, namely a transition from G to A. This symmetry condition is fulfilled by the Klein-Four-group, but not by the cyclic group on four states.

While the OSM approach can be extended to any number of states, its biological relevance becomes somewhat obscure when there is no corresponding group which is a power of $\mathbb{Z}^2$. In particular, there are four distinct Abelian groups for 20 states, but none fits a biologically meaningful binning of the 20 amino acids.

Competing interests

The authors declare no competing interests.

Authors' contributions

All authors contributed equally. All authors read and approved the final manuscript.

Acknowledgements

SK thanks Marston Conder for fruitful discussions on the group theoretical background and Jessica Leigh for enlightening discussions on biochemical and evolutionary binnings of amino acids. This work is financially supported by the Wiener Wissenschafts-, Forschungs- and Technologiefonds (WWTF). AvH also acknowledges the funding from the DFG Deep Metazoan Phylogeny project, SPP (HA1628/9) and the support from the Austrian GEN-AU project Bioinformatics Integration Network III.

Author details

[1]Department for Mathematics und Computer Science, Ernst-Moritz-Arndt-University Greifswald, Walther-Rathenau-Strasse 47, 17487 Greifswald, Germany. [2]Department of Statistics and School of Biological Sciences, University of Auckland, Private Bag 92019, Auckland, New Zealand. [3]Groningen Bioinformatics Centre, University of Groningen, Nijenborgh 7, 9747 AG Groningen, The Netherlands. [4]Center for Integrative Bioinformatics Vienna, Max F. Perutz Laboratories, University of Vienna, Medical University of Vienna, University of Veterinary Medicine Vienna, Dr. Bohr Gasse 9, A-1030, Vienna, Austria.

Received: 17 October 2011 Accepted: 10 December 2012
Published: 15 December 2012

References

1. Durbin R, Eddy SR, Krogh A, Mitchison G: *Biological sequence analysis - Probabilistic models of proteins and nucleic acids*. Cambridge: Cambridge University Press; 1998.
2. Mount DW: *Bioinformatics: Sequence and Genome Analysis*. New York: Cold Spring Harbor; 2004.
3. Semple C, Steel M: *Phylogenetics*. New York: Oxford University Press; 2003.
4. Nguyen MAT, Klaere S, von Haeseler A: **MISFITS: evaluating the goodness of fit between a phylogenetic model and an alignment.** *Mol Biol Evol* 2011, **28:**143–152.
5. Klaere S, Gesell T, von Haeseler A: **The impact of single substitutions on multiple sequence alignments.** *Philos T R Soc B* 2008, **363:**4041–4047.
6. Kimura M: **Estimation of Evolutionary Distances between Homologous Nucleotide Sequences.** *P Natl Acad Sci USA* 1981, **78:**454–458.
7. Fitch WM: **Toward defining the course of evolution: Minimum change for a specific tree topology.** *Syst Zool* 1971, **20:**406–416.
8. Humphreys JF: *A course in group theory*. New York: Oxford University Press; 1996.
9. Hendy M, Penny D, Steel M: **A discrete Fourier analysis for evolutionary trees.** *P Natl Acad Sci USA* 1994, **91:**3339–3343.
10. Bashford JD, Jarvis PD, Sumner JG, Steel MA: $U(1) \times U(1) \times U(1)$ **symmetry of the Kimura 3ST model and phylogenetic branching processes.** *J Phys A: Math Gen* 2004, **37:**L81—L89.
11. Bryant D: **Hadamard Phylogenetic Methods and the *n*-taxon process.** *Bull Math Biol* 2009, **71**(2):339–351.
12. Hendy MD, Penny D: **A framework for the quantitative study of evolutionary trees.** *Syst Zool* 1989, **38**(4):297–309.
13. MacLane S, Birkhoff G: *Algebra*. Chelsea: American Mathematical Society; 1999.
14. Bryant D: **Extending Tree Models to Split Networks.** In *Algebraic Statistics for Computational Biology*. Edited by Pachter L, Sturmfels B. Cambridge: Cambridge University Press; 2005.
15. Kimura M: **A Simple Method for Estimating Evolutionary Rates of Base Substitutions through Comparative Studies of Nucleotide Sequences.** *J Mol Evol* 1980, **16:**111–120.
16. Horn RA, Johnson CR: *Topics in matrix analysis*. New York: Oxford University Press; 1991.
17. Steeb WH, Hardy Y: *Matrix Calculus and Kronecker Product: A Practical Approach to Linear and Multilinear Algebra*. 2nd ed. Singapore: World Scientific Publishing; 2011.
18. Cayley A: **Desiderata and Suggestions: No. 2. The Theory of Groups: Graphical Representation.** *Am J Math* 1878, **1**(2):174–176.
19. Nguyen MAT, Gesell T, von Haeseler A: **ImOSM: Intermittent Evolution and Robustness of Phylogenetic Methods.** *Mol Biol Evol* 2012, **29**(2):663–673.
20. Horn RA, Johnson CR: *Matrix analysis*. Cambridge: Cambridge University Press; 1990.
21. Tamura K, Nei M: **Estimation of the number of nucleotide substitutions in the control region of mitochondrial DNA in humans and chimpanzees.** *Mol Biol Evol* 1993, **10**(3):512–526.
22. Sumner JG, Holland BR, Jarvis PD: **The Algebra of the General Markov Model on Phylogenetic Trees and Networks.** *Bull Math Biol* 2012, **74**(4):858–880.
23. Sumner JG, Jarvis PD, Fernandez-Sanchez J, Kaine B, Woodhams M, Holland BR: **Is the general time-reversible model bad for molecular phylogenetics?** *Syst Biol* 2012, **61**(6):1069–1074.

24. Keilen T: **Endliche Gruppen. Eine Einführung mit dem Ziel der Klassifikation von Gruppen kleiner Ordnung.** 2000. [http://www.mathematik.uni-kl.de/wwwagag/download/scripts/Endliche.Gruppen.pdf]
25. Kosiol C, Goldman N: **Different Versions of the Dayhoff Rate Matrix.** *Mol Biol Evol* 2005, **22**(2):193–199.
26. Susko E, Roger AJ: **On Reduced Amino Acid Alphabets for Phylogenetic Inference.** *Mol Biol Evol* 2007, **24**(9):2139–2150.

doi:10.1186/1748-7188-7-36
Cite this article as: Fischer *et al.*: On the group theoretical background of assigning stepwise mutations onto phylogenies. *Algorithms for Molecular Biology* 2012 **7**:36.